WEST

End of Result Set

☐ | Generate Collection | | Print |

DOCUMENT-IDENTIFIER: US 6185184 B1
TITLE: Directory server for providing dynamically assigned network protocol addresses

Abstract Text (1):
A communication protocol for establishing real-time, point-to-point communications between computer users over a computer network includes a directory server apparatus for providing the current dynamically assigned Internet Protocol addresses of client processes currently connected to the computer network. The server maintains a list of entries, each entry including the Internet Protocol address of a user currently connected to the network. In response to identification of one of the entries by a requesting client process, the server provides the corresponding Internet Protocol address of the entry to the requesting client process. In accordance with a second aspect of the present invention, the directory server monitors the status of client processes connected to the network via periodic notification from the client processes. The server dynamically modifies the time interval at which client processes notify the server, depending on the demand for server resources.

Parent Case Text (3):
To the extent that any matter contained herein is not already disclosed in the above-identified parent this application claims priority to U.S. provisional patent application Ser. No. 60/025,415 entitled Internet Telephony Apparatus and Method by Mattaway et al., filed Sep. 4, 1996, and U.S. provisional patent application Ser. No. 60/024,251 entitled System and Methods for Point-To-Point Communications Over a Computer Network, by Mattaway et al., filed Aug. 21, 1996.

Parent Case Text (6):
U.S. patent application Ser. No. 08/719,554, entitled Point-to-point Computer Network Communication Utility Utilizing Dynamically Assigned Network Protocol Addresses, now U.S. Pat. No. 6,131,121 by Mattaway et al.;

Parent Case Text (8):
U.S. patent application Ser. No. 08/719,891, entitled Method And Apparatus For Distribution And Presentation Of Multimedia Data Over A Computer Network, now pending by Mattaway et al.;

Parent Case Text (9):
U.S. patent application Ser. No. 08/719,898, entitled Method And Apparatus For Providing Caller Identification Based Out-going Messages In A Computer Telephony Environment, now pending by Mattaway et al.;

Brief Summary Text (2):
The present invention relates, in general, to data processing systems, and more specifically, to a method and apparatus for facilitating audio communications over computer networks.

Brief Summary Text (4):
The increased popularity of on-line services such as AMERICA ONLINE.TM., COMPUSERVE.RTM., and other services such as Internet gateways have spurred applications to provide multimedia, including video and voice clips, to online

users. An example of an online voice clip application is VOICE E-MAIL FOR WINCIM and VOICE E-MAIL FOR AMERICA ONLINE.TM., available from Bonzi Software, as described in "Simple Utilities Send Voice E-Mail Online", MULTIMEDIA WORLD, VOL. 2, NO. 9, August 1995, p. 52. Using such Voice E-Mail software, a user may create an audio message to be sent to a predetermined E-mail address specified by the user.

Brief Summary Text (5):
Generally, devices interfacing to the Internet and other online services may communicate with each other upon establishing respective device addresses. One type of device address is the Internet Protocol (IP) address, which acts as a pointer to the device associated with the IP address. A typical device may have a Serial Line Internet Protocol or Point-to-Point Protocol (SLIP/PPP) account with a permanent IP address for receiving E-mail, voicemail, and the like over the Internet. E-mail and voicemail is generally intended to convey text, audio, etc., with any routing information such as an IP address and routing headers generally being considered an artifact of the communication, or even gibberish to the recipient.

Brief Summary Text (8):
The ability to locate, users having temporary or dynamically assigned Internet Protocol address has been difficult without the user manually initiating the communication. Accordingly, spontaneous, real-time communications with such users over computer networks have been impractical. Further, even if the dynamically assigned Internet Protocol address of a user was obtained, it is difficult to monitor whether the client is still coupled to the computer network and whether the dynamically assigned Internet Protocol address is still valid. As such, computer users typically have little more information about a party than the E-address, name, alias, or telephone number of the party, with no knowledge of whether the party is currently connected to the Internet or the dynamically assigned Internet Protocol address of the party.

Brief Summary Text (9):
Accordingly, a need exists for a way to determine whether computer users are actively connected to a computer network.

Brief Summary Text (10):
A further need exists for a way to obtain the dynamically assigned Internet Protocol address of a user; having on-line status with respect to a computer network, particularly the Internet.

Brief Summary Text (13):
The above deficiencies in the prior art and the previously described needs are fulfilled by the present invention which provides, a directory server utility for providing the dynamically assigned network protocol addresses of client processes currently coupled to the computer network. Accordingly to one embodiment of the present invention, a method of locating users having dynamically assigned network protocol addresses comprises the steps of maintaining a compilation of entries, each entry comprising a network protocol address of a client process connected to the computer network, and, in response to identification of one of the entries by a requesting client process, providing the network protocol address of the identified entry to the requesting client process.

Brief Summary Text (14):
According to another embodiment of the present invention, a computer program product, for use with a computer server operatively coupled over a computer network to one or more client processes, comprises a computer useable medium having program code means for maintaining a compilation of entries, each entry comprising the network protocol address of a client process. Program code means, responsive to identification of one of the entries by a requesting client process, provide the network protocol address associated with the entry to the requesting client process.

Brief Summary Text (15):
In accordance with another aspect of the present invention, a server monitors the status of multiple client processes actively connected to the computer network with a method comprising the steps of receiving notification from a client process that

the client process is active, and, determining that the client process is inactive if a subsequent notification is not received within a predetermined time interval. The method further contemplates the steps of modifying the predetermined time interval depending on utilization of server resources.

Drawing Description Text (12):
FIG. 10 illustrates schematically a computer network over which the present invention may be utilized;

Drawing Description Text (15):
FIGS. 13A-B are schematic block diagrams of the elements comprising the inventive computer network telephony mechanism of the present invention;

Detailed Description Text (2):
Referring now in specific detail to the drawings, with like reference numerals identifying similar or identical elements, as shown in FIG. 1, the present disclosure describes a point-to-point network protocol and system 10 for using such a protocol.

Detailed Description Text (3):
In an exemplary embodiment, the system 10 includes a first processing unit 12 for sending at least a voice signal from a first user to a second user. The first processing unit 12 includes a processor 14, a memory 16, an input device 18, and an output device 20. The output device 20 includes at least one modem capable of, for example, 14.4Kilobit-per-second communications and operatively connected via wired and/or wireless communication connections to the Internet or other computer networks such as an Intranet, i.e., a private computer network. One skilled in the art would understand that the input device 18 may be implemented at least in part by the modem of the output device 20 to allow input signals from the communication connections to be received. The second processing unit 22 may have a processor, memory, and input and output devices, including at least one modem and associated communication connections, as described above for the first processing unit 12. In an exemplary embodiment, each of the processing units 12, 22 may execute the WEBPHONE.RTM. Internet telephony application available from NetSpeak Corporation, Boca Raton, Fla., which is capable of performing the disclosed point-to-point Internet protocol and system 10, as described herein.

Detailed Description Text (7):
In the illustrative embodiment, the POP protocol is utilized to retrieve E-mail messages from mail server 28 while the SMTP protocol is used to submit E-mail message to Internet 24.

Detailed Description Text (8):
The first processing unit 12 may operate the disclosed point-to-point Internet protocol by a computer program described hereinbelow in conjunction with FIG. 6, which may be implemented from compiled and/or interpreted source code in the C++ programming language and which may be downloaded to the first processing unit 12 from an external computer. The operating computer program may be stored in the memory 16, which may include about 8 MB RAM and/or a hard or fixed drive having about 8 MB of available memory. Alternatively, the source code may be implemented in the first processing unit 12 as firmware, as an erasable read only memory (EPROM), etc. It is understood that one skilled in the art would be able to use programming languages other than C++ to implement the disclosed point-to-point network protocol and system 10.

Detailed Description Text (15):
For callee (or called) processing units with fixed IP addresses, the caller (or calling) processing unit may open a "socket", i.e. a file handle or address indicating where data is to be sent, and transmit a command to establish communication with the callee utilizing, for example, datagram services such as Internet Standard network layering as well as transport layering, which may include a Transport Control Protocol (TCP) or a User Datagram Protocol (UDP) on top of the IP. Typically, a processing unit having a fixed IP address may maintain at least one open socket and a called processing unit waits for a command to assign the open socket to the incoming signal. If all lines are in use, the callee processing unit

sends a BUSY signal or message to the caller processing unit. As shown in FIG. 1, the disclosed point-to-point Internet protocol and system 10 operate when a callee processing unit does not have a fixed or predetermined IP address. In the exemplary embodiment and without loss of generality, the first processing unit 12 is the caller processing unit and the second processing unit 22 is the callee processing unit. When either of processing units 12, 22 logs on to the Internet via a dial-up connection, the respective unit is provided a dynamically allocated IP address by an Internet service provider.

Detailed Description Text (20):
If the callee is not oh-line when the connection server 26 determines the callee's status, the connection server 26 sends an OFF-LINE signal or message to the first processing unit 12. The first processing unit 12 may also display a message such as "Called Party Off-Line" to the first user.

Detailed Description Text (21):
When a user logs off or goes off-line from the Internet 24, the connection server 26 updates the status of the user in the database 34; for example, by removing the user's information, or by flagging the user as being off-line. The connection server 26 may be instructed to update the user's information in the database 34 by an off-line message, such as a data packet, sent automatically from the processing unit of the user prior to being disconnected from the connection server 26. Accordingly, an off-line user is effectively disabled from making and/or receiving point-to-point Internet communications.

Detailed Description Text (22):
As shown in FIGS. 2-4, the disclosed secondary point-to-point Internet protocol may be used as an alternative to the primary point-to-point Internet protocol described above, for example, if the connection server 26 is non-responsive, unreachable, inoperative, and/or unable to perform the primary point-to-point Internet protocol, as a non-responsive condition. Alternatively, the disclosed secondary point-to-point Internet protocol may be used independent of the primary point-to-point Internet protocol. In the disclosed secondary point-to-point Internet protocol, the first processing unit 12 sends a message via E-mail over the Internet 24 to the mail server 28. The E-mail including the message may have, for example, the subject

Detailed Description Text (24):
The following E-mail messages are transmitted to a remote users post office protocol server via simple mail transport protocol using MIME by the event manager, as explained hereinafter.

Detailed Description Text (26):
The following E-mail messages are received from a local WebPhone users POP server via the POP protocol using MIME by the event manager, as explained hereinafter.

Detailed Description Text (27):
As described above, the first processing unit 12 may send the message in response to an unsuccessful attempt to perform the primary point-to-point Internet protocol. Alternatively, the first processing unit 12 may send the message in response to the first user initiating a SEND command or the like.

Detailed Description Text (28):
After the message via E-mail is sent, the first processing unit 12 opens a socket and waits to detect a response from the second processing unit 22. A timeout timer, such as timer 32, may be set by the first processing unit 12, in a manner known in the art, to wait for a predetermined duration to receive a signal. The processor 14 of the first processing unit 12 may cause the output device 20 to output a Ring signal to the user, such as an audible ringing sound, about every 3 seconds. For example, the processor 14 may output a *.wav file, which may be labeled RING.WAV, which is processed by the output device 20 to output an audible ringing sound.

Detailed Description Text (29):
Second processing unit 22 polls mail server 28 at an interval, for example, once a minute, to check for incoming E-mail. Generally, second processing unit 22 checks the messages stored on mail server 28 at regular intervals to wait for and detect

incoming E-mail indicating a <u>message</u> from first processing unit 12.

Detailed Description Text (31):
The E-mail may subsequently be retrieved by processing unit 22 on behalf of the user with another standard protocol, for example POP 3. The actual IP address utilized by the user's processing unit is immaterial to the retrieval of E-mail, as the mail server 28 can, for example, be polled or queried from any point on the <u>network</u>.

Detailed Description Text (33):
Point-to-point communication may then be established by the processing unit 22 processing the E-mail signal to extract the <u>message,</u> including the IP address of the first processing unit 12 and the session number. The second processing unit 22 may then open a socket and generate a response signal, which includes the temporary IP address of the second processing unit 22 as well as the session number of the first processing unit.

Detailed Description Text (36):
In the exemplary embodiment, a first user operating the first processing unit 12 is not required to be notified by the first processing unit 12 that an E-mail is being generated and sent to establish the point-to-point link with the second user at the second processing unit 22. Similarly, the second user is not required to be notified by the second processing unit 22 that an E-mail has been received and/or a temporary IP address is associated with the second processing unit 22. The processing units 12, 22 may perform the disclosed point-to-point Internet protocol automatically upon initiation of the point-to-point communication command by the first user without displaying the E-mail interactions to either user. Accordingly, the disclosed point-to-point Internet protocol may be transparent to the users. Alternatively, either of the first and second users may receive, for example, a brief <u>message</u> of "CONNECTION IN PROGRESS" or the like on a display of the respective output device of the processing units 12, 22.

Detailed Description Text (37):
After the initiation of either the primary or the secondary point-to-point Internet protocols described above in conjunction with FIGS. 1-2, the point-to-point communication link over the Internet 24 may be established as shown in FIGS. 3-4 in a manner known in the art. For example, referring to FIG. 3, upon receiving the signal from the second processing unit 22, the first processing unit 12 extracts the IP address of the second processing unit 22 and the session number, and the session number sent from the second processing unit 22 is then checked with the session number originally sent from the first processing unit 12 in the <u>message</u> as E-mail. If the session numbers sent and received by the processing unit 12 match, then the first processing unit 12 sends a signal directly over the Internet 24 to the second processing unit 22; i.e. using the IP address of the second processing unit 22 provided to the first processing unit 12 in the signal.

Detailed Description Text (42):
As shown in FIG. 5, a first display screen 36 includes a status area 38 for indicating, for example, a called user by name and/or by IP address or telephone number; a current function such as C2; a current time; a current operating status such as "IN USE", and other control icons such as a down arrow icon 40 for scrolling down a list of parties on a current conference line. The operating status may include such annunciators as "IN USE, " "IDLE," "BUSY," "NO ANSWER," "OFFLINE," "CALL," "DIALING," "<u>MESSAGES,</u>" and "SPEEDDIAL."

Detailed Description Text (51):
Referring to FIG. 7, the disclosed point-to-point Internet protocol and system 10 is illustrated. First processing unit 12 initiates the point-to-point Internet protocol in step 56 by sending a query from the first processing unit 12 to the connection server 26. If connection server 26 is operative to perform the point-to-point Internet protocol, in step 58, first processing unit 12 receives an on-line status signal from the connection server 26, such signal may include the IP address of the callee or a "Callee Off-Line" <u>message</u>. Next, first processing unit 12 performs the primary point-to-point Internet protocol in step 60, which may include receiving, at the first processing unit 12, the IP address of the callee if the callee is active and on-line. Alternatively, processing unit 60 may initiate and perform the

secondary point-to-point Internet protocol in step 62, if connection server 26 is not operable.

Detailed Description Text (54):
The primary and secondary point-to-point Internet protocols previously described enable users to establish real-time direct communication links over the Internet or other computer networks without the need for any interaction with connection server 26, the connection server providing only directory and information related services.

Detailed Description Text (55):
FIG. 10 illustrates an exemplary computer network 1000 over which the invention may operate. A first processing unit 1012 is coupled to a computer network, illustrated here as the Internet 1010, through an Internet service provider 1014 Similarly, a second processing unit 1022 is coupled to Internet 1010 through Internet service provider 1018. The inventive directory server 1020 is similarly coupled to Internet 1010 through Internet service provider 1026. Directory server 1020 further comprises a connection server 1022 and information server 1024, as will be explained hereinafter. The first processing unit 1012, second processing unit 1022 and directory server 1020 are operatively coupled to each other via the Internet 1010. It will be obvious to those reasonably skilled in the art that network 1000 is not restricted to implementation over the Internet 1010 but may comprise other network configurations such as a local area network (LAN), a wide area network (WAN), a global area network or any number of private networks currently referred to as an Intranet. Such networks may be implemented with any number of hardware and software components, transmission media and network protocols.

Detailed Description Text (60):
User input to computer system 100 may be provided by a number of devices. For example, a keyboard 1156 and mouse 1157 are connected to bus 1130 by controller 1155. An audio transducer 1196, which may act as both a microphone and a speaker, is connected to bus 1130 by audio controller 1197, as illustrated. It will be obvious to those reasonably skilled in the art that other input devices, such as a pen and/or tablet may be connected to bus 1130 with an appropriate controller and software, as required. DMA controller 1160 is provided for performing direct memory access to RAM 1110. A visual display is generated by video controller 1165 which controls video display 1170. Computer system 1100 also includes a communications adaptor 1190 which allows the system to be interconnected to a network such as a local area network (LAN), a wide area network (WAN), or the Internet, schematically illustrated by transmission medium 1191 and network 1195.

Detailed Description Text (63):
FIG. 12 illustrates schematically an audio sound card 1200 which may be used to implement audio controller 1197 of FIG. 11. Specifically, sound card 1200 may comprise, in the exemplary embodiment, an analog-to-digital (A/D) converter 1212, an input buffer 1216, a digital signal processor (DSP) 1222, ROM 1224, RAM 1226, an output buffer 1220, and an analog-to-digital (D/A) converter 1218, all of which may be interconnected over a bus 1210. Bus 1210 is in turn coupled to a bus interface 1228 which, in turn, is coupled to bus controller 1125 of computer system 1100 of FIG. 11.

Detailed Description Text (64):
As illustrated in FIG. 12, A/D converter 1212 is coupled to audio transducer 1214 which is typically a microphone. Conversely, D/A converter 1218 is coupled to audio transducer 1230, typically a speaker. It will be obvious to those reasonably skilled in the art that audio transducers 1214 and 1230, may be combined into a single element which serves as both a transmitter and receiver of audio signal.

Detailed Description Text (65):
In operation, A/D converter 1212 samples the audio signals supplied to it by transducer 1214 and stores the digital samples in buffer 1216. The digital sampling occurs under control of a program typically stored in ROM 1224, or, alternatively, under the control of digital signal processor 1222. The digital samples stored in input buffer 1216 are forwarded periodically, typically when the buffer reaches near capacity, over bus 1210 to bus 1130 of FIG. 11, for further processing by computer

system 1100. The device driver for audio sound card 1200 generates system interrupts which will cause the digital samples stored in input buffer 1216 to be retrieved for processing. In the exemplary embodiment, the digital samples are uncompressed as supplied to computer system 1100. However, compression of the digital samples may occur using DSP 1222 executing an appropriate compression algorithm, if desired.

Detailed Description Text (66):
Digital audio samples from computer system 1100 are also be converted to analog signals by sound card 1200. The digital samples are supplied to bus 1210 and temporarily stored into output buffer 1220. The digital samples are then converted by D/A converter 1218 into an analog signals which are then supplied to audio transducer 1230, i.e., a speaker, or to further amplification and processing devices.

Detailed Description Text (69):
In the exemplary embodiment of the present invention, each of first processing unit 1012 and second processing unit 1022 of FIG. 10 are executing a software application capable of enabling point-to-point communication over network 1000, such as an Internet telephone application. One such application suitable for use with the present invention is the WebPhone Version 1.0 or higher, software, hereafter referred as the "WebPhone," commercially available from NetSpeak Corporation, Boca Raton, Fla. A description of the architecture and operation of the WebPhone is provided herein with reference to FIGS. 5-6, 13A-B and 14. An extensive detailed description of the architecture, application program interface, graphic user interface, and operation of the WebPhone can be found in copending U.S. patent application Ser. No. 08/719,554 entitled "Point-to-Point Computer Network Communication Utility Utilizing Dynamically Assigned Internet Protocol Addresses" by Mattaway et al. filed on an even date herewith and commonly assigned, the complete subject matter of which is incorporated herein by reference.

Detailed Description Text (70):
Referring to FIGS. 13A-B, schematic block diagrams of the WebPhone architecture are illustrated. The WebPhone is an end-user software application which enables users to send real-time audio data to other WebPhone users over the Internet or any public or private TCP/IP based computer networks. The WebPhone application and architecture may be designed to run on any number of operating systems or computer architectures. In the illustrative embodiment, the WebPhone application is implemented as a Windows compatible application executable on an IBM PC architecture or a clone thereof.

Detailed Description Text (71):
Referring to FIG. 13A, the WebPhone 1300 comprises a set of object modules, written in a programming language such as C++, which work together in a concerted fashion to provide real-time, multitasking, network-based media transmission and reception. WebPhone 1390 comprises a graphic user interface (GUI) 1310, a user interface (UI) 1312, an event manager 1314, a media engine 1316, a database dynamic link library 1318, one or more audio compression/decompression (codecs) 1320, an audio manager 1324, a WebPhone application program interface (API) 1326, and a network interface 1322.

Detailed Description Text (75):
WebPhone media engine 1316 manages the allocation of associated resources to provide a multitasking environment and controls the flow of real-time data streams, e.g., conversations, outgoing messages, etc., and non-real-time data streams, e.g., voice mail, graphic images, files, etc., to and from a user network connection. The objects representing tasks are created by event manager 1314, thereby freeing media engine 1316 to manage resource routing. Specifically, the media engine routes data streams from sources such as a microphone, file or network socket, to destinations such as speaker, destination file or other network socket. To perform such routing functions the media engine interfaces with the WebPhone API 1326 to control communication with other processes, and further communicates with audio manager 1324 to communicate with the system input/output apparatus, such as sound card 1200 of FIG. 12. Media engine 1314 may be designed to employ heuristic methods to sense and efficiently utilize available bandwidth to achieve timely and accurate delivery of all data streams, both real-time and non-real-time.

Detailed Description Text (76):
Media engine 1316, further interacts with WebPhone codec 1320 to achieve compression and decompression of audio data streams. Codec 1320 provides coding of digital samples from the sound card 1200 of FIG. 12 into a compressed format more suitable for transmission over a computer network. Codec 1320 further provides decoding of a compressed signal prior to its submission to sound card 1200 for subsequent conversion to an audible analog signal. In the exemplary embodiment, WebPhone codec 1320 is implemented in a modular fashion so that codecs may be replaced and updated with newer, more efficient compression/decompression algorithms via the Plug and Play protocol. A codec suitable for use with the present invention is the True Speech codec, version 8.5, commercially available from the DSP Group, Inc., Santa Clara, Calif. The True Speech codec is an enhanced linear predicative coding algorithm, specifically designed to efficiently encode and decode human speech data. The True Speech codec samples the digital sample stream from sound card 1200, and, using a look-up table-based algorithm, tries to predict the value of the next data sample in the digital data stream based on the history of prior data sample values. The compressed data stream comprises a combination of identifiers of the predicted sample values, as well as error values used to correct the predictive values. Accordingly, the a mount of digital data actually transmitted to represent the audio signal is significantly reduced in comparison to transmission of the actual data samples generated by sound card 1200. The True Speech codec provides temporal, frequency domain compression of the digital data representing the audio signal.

Detailed Description Text (78):
The WebPhone API 1326 enables the WebPhone to communicate with other WebPhones, connection and directory assistance servers, Internet gateway servers, credit processing servers, database access servers and other client processes implementing the WebPhone API. As illustrated in FIG. 13B, the WebPhone API utilizes sockets, i.e., a file handle or address indicating where data is to be sent, allowing WebPhone API enabled processes to reside on the same computer, on a local area network, on a wide area network, or over the Internet. A process 1328 communicates with the WebPhone API 1326 through a plurality of sockets 1322. The sockets 1322 are accessible by network 1330 through a number of protocols including Internet Protocol (IP) 1332, Transmission Control Protocol (TCP) 1334, Real-Time Protocol (RTP) 1336 and User Datagram Protocol (UDP) 1338. The WebPhone API provides remote command control of WebPhones and servers via the TCP. WebPhone API 1326 transfers real-time and streamed audio via the UDP protocol and real-time audio and video data via the UDP and RTP protocols. The WebPhone API utilizes TCP to transfer data of different types, i.e., file, image, graphics, etc. as well as to transfer streamline video and other multimedia data types, such as Java developed by Sun MicroSystems, Mountain View, Calif. In addition, the WebPhone API provides user definable commands and data types.

Detailed Description Text (81):
Having described the architecture of the WebPhone software which enables the first and second processing units to establish point-to-point communication over a network, a discussion of the global connection/information server is appropriate.

Detailed Description Text (82):
Referring to FIG. 15A, a network diagram, similar to that shown in FIG. 10, is illustrated, including a schematic diagram of the global server 1500 and the various devices operatively coupling server 1500 to the Internet 1530. A first processing unit executing the WebPhone application, hereafter referred to as WebPhone 1536, is coupled to Internet 1530 through an Internet service provider 1532. Similarly, a second processing unit executing the WebPhone application, referred to as WebPhone 1538, is coupled to the Internet 1530 by an Internet service provider 1534. Global server 1500 is coupled to Internet 1530 by an Internet service provider 1528, a CSU/DSU 1526, a router 1524, and a fire wall server 1522. In the illustrative embodiment, fire wall server 1522 and global server 1500 are connected through a local area network 1520. Network 1520 may be implemented with an Ethernet or other suitable transport for TCP/IP communications. However, as will be obvious to those recently skilled in the arts, server 1500 may be connected directly to fire wall server 1522.

Detailed Description Text (83):

In the illustrative embodiment, firewall server 1522 is a single firewall mechanism which protects unauthorized access from network 1530 into global server 1500. Firewall server 1522 may be implemented on a work station, such as a SPARC 5 or SPARC 20 server from Sun MicroSystems, executing a commercially available firewall software application such as Raptor, available from Raptor Systems. Essentially, the firewall server prevents unauthorized access into global server 1500 and thereby prevents destruction of any of the information contained therein by checking the source of requests for information to global server 1500.

Detailed Description Text (85):
CSU/DSU 1526 (Channel Send Unit/Data Send Unit) functions as a sophisticated modem, converting network data to high speed serial data for transfer over a T1 or T3 line. Such high speed data is connected to another CSU/DSU, typically at the telephone company over the T1 or T3 line. An apparatus suitable for use in implementing CSU/DSU 1526 in the present invention is the AT&T Paradigm by AT&T Laboratories, Murray Hill, N.J.

Detailed Description Text (88):
Connection server 1512 provides a directory information service to WebPhone client processes currently on-line with respect to the computer network. Connection server 1512 behaves like a virtual machine within global server 1500 and interacts with database 1516 through database server 1518 and with network interface card 1540 through the WebPhone API. The basic function of connection server 1512 is to provide a one-to-one mapping between an identifier of a WebPhone client process, such as a E-mail address, and the current IP address, dynamic or fixed, associated with that WebPhone client process.

Detailed Description Text (92):
Information server 1514 provides an interface between requests from WebPhone client processes and database 1516. Information server 1514 includes code written to extract the search criteria from an packet and supply the search criteria to the database search engine of database 1516 using the ODBC standard. In particular, referring to FIG. 16B, a flow chart illustrating the basic process steps used by information server 1514 in performing information/directory service functions in accordance with the present invention is illustrated. The coding of the process steps of the flow chart into instructions suitable for execution by global server 1500 will be understood by those having ordinary skill in the art of programming. Information server 1514 remains idle until an packet is received from a WebPhone client process, as illustrated by decisional step 1630. Next, information server 1514 extracts the data elements defined within the packet and supplies them to database server 1518 which, in turn, forward them to database 1516, as represented by the process step 1634 and 1636. The search engine contained within database 1516 performs the search and supplies to information server 1514 all client records meeting the search criteria specified in the packet, or a message indicating that no records were found. Next, information server 1514 transmits a packet to the WebPhone client process indicating the number of records satisfying the search criteria, as indicated by process step 1638. The WebPhone client may wish to receive all records satisfying the search criteria, or, if the number is excessively large, may desire to further refine the search by transmitting a packet to information server 1514 and defining new search parameters to be sent with a subsequent packet. If a packet is received by information server 1514, the process will return to an idle state, as illustrated by decisional block 1640. If no packet was received, information server 1514 will transmit one or more packets to the requesting WebPhone client until all records have been received by the WebPhone client, as illustrated by process step 1642. Information server 1514 will return to an idle state awaiting another packet, as illustrated in FIG. 16B. A description of the packets comprising the WebPhone protocol is illustrated in Tables 7-8 and a diagram illustrating the packet transfer sequence defined in FIG. 17A-B.

Detailed Description Text (93):
Network interface card 1540 interfaces with connection server 1512, information 1514, and database server 1518 using the WebPhone API definition, as described herein, and the Windows Sockets 1.1 Protocol, or, in a Unix-based operating system, Berkeley Sockets Network API. Network interface card 1514 may comprise, in illustrative embodiment, an Ethernet card capable of transmitting data at rates of

100 Mbps or greater, such cards being commercially available through a number of different vendors.

Detailed Description Text (98):
The On-line table 1516B provides a dynamic list of those clients from 1516A who are currently On-line, as well as their current Internet protocol address. On-line Table 1516B comprises a plurality of records each of which may have the fields and data types illustrated in Table 2. The record entries of On-line table 1516B are used by connection server 1512 and information server 1514, as explained hereinafter, to provide a directory of those WebPhone client processes currently having on-line status with respect to the computer network.

Detailed Description Text (99):
The WebBoard.TM. is a virtual multimedia billboard which is transmitted as a series of multimedia data files to WebPhone client processes while the WebPhone application is activated. An extensive description of the WebBoard utility and its operation can be found in copending U.S. patent application Ser. No. 08/719,891 entitled Method and Apparatus for Distribution of Multimedia Data Over a Computer Network by Mattaway et al., commonly assigned, the subject matter of which is incorporated herein by reference.

Detailed Description Text (100):
A number of tables are associated with the WebBoard functionality including WebBoard table 1516C, a WebBoard configuration table 1516D, and a WebBoard source table 1516E. WebBoard table 1516C includes a plurality of records each describing a specific WebBoard and having the field and data types illustrated in Table 3. The "id" field of Table 3 provides a unique identification number for the WebBoard file. The "imageType" field defines the video format of the image such as JPEG, TIF, GIF, etc. The "audio" field defines the nature of the audio file, e.g. a.wav file or a MIDI file, while the "audioType" field defines the codec, if any, used to compress/decompress the audio file. The "hits" field defines the number of times the WebBoard has been selected by WebPhone clients, while the "hits profile" field defines the file name of the file identifying those WebPhone clients generating hits to the subject WebBoard.

Detailed Description Text (104):
Database server 1518 serves as the interface between database 1516 and connection server 1512 and information server 1514. Specifically, connection server 1512 and information server 1514 communicate with database engine 1518 through application program interfaces embedded in the code implementation of both the connection server and the information server. Database server 1518 communicates with database 1516, in the illustrative embodiment, using the open database connectivity (ODBC) standard, developed by Microsoft Corporation, Redmond, Wash. Database server 1518 functions to supply structured database queries to database 1516 and to supply the results therefrom to connection server 1514 and information server 1512. In the illustrative embodiment, database server 1518 may be implemented as a "virtual machine" executing on global server 1500, or, alternatively, may be implemented on a separate computer system such as a DEC Alpha 4100 Workstation executing DEC Unix operating system, both available from Digital Equipment Corporation, Maynard, Mass. Database server 1518 communicates with network interface card 1518 using the WebPhone Application Program Interface described herein.

Detailed Description Text (105):
Global Server Network

Detailed Description Text (106):
In the illustrative embodiment, global server 1500 is implemented as a single server apparatus on which a plurality of "virtual machines" execute simultaneously. However, it will be obvious to those reasonably skilled in the art that a plurality of separate servers, one dedicated to each of connection server 1512, information server 1514, and database server 1518 may be interconnected to database 1516 and to each other using a local area network, to form a composite "virtual" global server, as illustrated by FIG. 15B, the construction of the system illustrated in FIG. 15B being within the knowledge of those reasonably skilled in the art in light of the descriptions contained herein.

Detailed Description Text (107):
It is further contemplated within the present invention that more than one global
server 1500 may be utilized, as illustrated by FIG. 15C. In this implementation,
multiple global servers 1500A-D are maintained for fault tolerant load sharing, each
one performing the above-described connection server, information server and
database server processes. Each of global servers 1500A-D are connected to the
Internet via a separate T1 or T3 connection to different Internet service providers,
and are synchronized with each other via database server replication. In such an
embodiment, multiple global servers may be located in close proximity or in
geographically disparate locations. In such an embodiment, the WebPhone application
is provided with the network address information of each global server 1500A-D. In
the event that any one of the global servers initially contacted is nonresponsive
the WebPhone application will attempt connection to one or more of the remaining
global servers to obtain directory and information services.

Detailed Description Text (108):
Further, in an implementation with multiple global servers, if the initially
contacted global server is unable to accommodate a WebPhone client request, or, is
not geographically convenient, the global server can provide the network address of
another global server capable of servicing the WebPhone client's request or which is
logically more convenient. This process may occur during the initial log-in of the
WebPhone client process, as described with references to messages 1-5 of FIG. 17A.

Detailed Description Text (109):
As previously described, if none of the global servers are available, the WebPhone
application can rely on the secondary Internet Protocol technique in which a
WebPhone client process sends its current dynamically assigned Internet Protocol
address to a prospective WebPhone callee through an E-mail message, as described
herein.

Detailed Description Text (112):
FIG. 17A illustrates a schematic block diagram of a packet transfer sequence between
a pair of WebPhone client processes and the global server, in accordance with the
present invention. Each WebPhone application, also referred to as a WebPhone client
process, connects to global server 1500 upon start up to inform global server 1500
that the WebPhone client process is on-line and available to make and/or receive
calls. Specifically, as illustrated in FIG. 17A, WebPhone 1536 opens a socket to the
global server 1500 and transmits an packet from WebPhone 1536 to Global server 1500,
as illustrated by message 1 and FIG. 17A. The packet may have the format and data
illustrated in Table 6, and additional Feature bits which define the functionality
of the WebPhone application, as explained in greater detail hereinafter. In
response, connection server 1512 and information server 1514 of global server 1500
use the information contained in the packet to update the status of database 1516.
In the event that the WebPhone client process is logging on for the first time,
global server 1500 returns to the WebPhone 1536 a packet, as illustrated by message
2 of FIG. 17A. The packet includes the elements as defined in Table 9. In response,
WebPhone 1536 returns a packet as illustrated by message 3 of FIG. 17A. The packet
contains the data elements defined in Table 8. Connection server 1512 and
information server 1514 of global server 1500 utilize the data in the packet to
update database 1516. Specifically, information server 1514 utilizes such data to
create a record in client table 1516A representing WebPhone 1536. Next, global
server 1500 transmits to WebPhone 1536 a packet, as illustrated by message 4 of
FIGS. 17A. The packet contains the data described in Table 7 plus Feature bits, as
described hereinafter. The packet returned to WebPhone 1536 enables certain
functions within the WebPhone architecture based on predetermined criteria, for
example, whether the user has paid for the product, or which version of the product
the user possesses. Following the packet, global server 1500 further transmits an
packet, as illustrated by message 5 of FIG. 17A. Prior to transmission of the
packet, connection server 1514 updates database 1516, specifically On-line table
1516B to indicate that WebPhone 1536 is on-line with respect to the computer
network. Upon receiving the packet, WebPhone 1536 closes the socket to global server
1500.

Detailed Description Text (113):

In the event WebPhone 1536 had previously registered with global server 1500, only messages 1 and 5 are required to establish WebPhone 1536 as being on-line. If WebPhone 1536 had new user information to supply to global server 1500, then packet sequence illustrated by messages 3 and 4 would occur.

Detailed Description Text (114):
Although the packet sequence illustrated by messages 1-5 is described with reference to WebPhone 1536, WebPhone 1538 interacts in a similar manner with global server 1500 to establish on-line status. No further interaction occurs between the respective WebPhone client processes and the global server unless the WebPhones require directory or search assistance about a prospective callee.

Detailed Description Text (115):
In one calling scenario, a WebPhone user knows the E-mail address of another WebPhone user to which he/she wishes to establish a point-to-point communication, however, the current dynamically assigned Internet protocol address of the callee is unknown to the caller. In this scenario, the user of WebPhone 1536 requests assistance from global server 1500 to obtain the current dynamically assigned Internet Protocol address of the prospective callee WebPhone. First, the user of WebPhone 1536 specifies the callee by entering all or part of the callee party's name or alias in the party name field area of the graphic user interface. If the party is not in the WebPhone user's local directory, the IP address or E-mail address of the callee WebPhone may be entered into the number field area of the graphic user interface, followed by activation of the send button or icon on the graphic user interface. As a result, WebPhone 1536 opens a socket to global server 1500 and transmits a packet having the format described in Table 6. Connection server 1512 of global server 1500 utilizes the value of the E-mail address specified in the packet to perform a one-to-one mapping in the online table 1516B to determine the current Internet Protocol address of the indicated callee, as illustrated by the flowchart of FIG. 15A. Once this mapping is performed, the server 1500 transmits to WebPhone 1536 a packet, as indicated by message 7A of FIG. 17A. The packet has the format and content as illustrated in Table 6 and includes the IP address of the callee as Well as information such as an error code to indicate that no WebPhone application is associated with that callee. Alternatively, if the selected callee is off line, global server 1500 transmits to WebPhone 1536 an packet to indicate that the desired party is not on-line, as illustrated by message 7B of FIG. 17A. Following the receipt of either a or an packet by WebPhone 1536, the socket to global server 1500 opened by WebPhone 1536 is closed.

Detailed Description Text (116):
If the current Internet Protocol address of the callee was returned from global server 1500, the packet transmission sequence illustrated between WebPhones 1536 and 1538 of FIG. 17A transpires. Whether a calling WebPhone knows the Internet Protocol address of the callee WebPhone, as in the case of a fixed Internet Protocol address, or obtains the Internet Protocol address from global server 1500, as previously described, the calling sequence to establish a call occurs as follows. WebPhone 1536 opens a socket to WebPhone 1538. Next, WebPhone 1536 transmits to WebPhone 1538 a packet as illustrated by message 8 of FIG. 16A. The packet has the format illustrated in Table 6 and may, optionally, include information identifying the compression/decompression (codec) used by the caller WebPhone. In response to the packet, WebPhone 1538 may return with a number of different packets, as illustrated by messages 9A-D. First, callee WebPhone 1538 may respond to caller WebPhone 1538 with a packet, as illustrated by message 9A, indicating that the callee WebPhone does not wish to be disturbed, e.g. total call blocking, or, that the callee WebPhone does not wish to talk to caller WebPhone, e.g. party specific or group specific call blocking. In the event of party or group specific call blocking, the user information contained within the packet of message 9A is compared by the caller WebPhone application to a predefined list of WebPhone user information profiles which the callee does not wish to converse, such list having been predefined by the callee in the WebPhone user's personal directory, as explained hereinafter. Upon receiving the packet the caller WebPhone annunciates the result to the user and the socket to the callee WebPhone is closed.

Detailed Description Text (117):
Alternatively, callee WebPhone 1538 may return a packet, as illustrated by message

9B of FIG. 17A. The packet indicates that the callee WebPhone is currently utilizing all available lines within its WebPhone application.

Detailed Description Text (118):
A further possible response from callee WebPhone 1538 is to issue an packet, as illustrated by message 9C of FIG. 17A. The packet includes data indicating whether the machine is capable of receiving voice mail type messages as described in greater detail in copending U.S. patent application Ser. No. 08/719,898 entitled "Method and Apparatus for Providing Caller Identification Based Out-Going Messages in a Computer Telephony Environment," by Mattaway et al., commonly assigned and incorporated herein by reference.

Detailed Description Text (119):
The preferred response by callee WebPhone 1538 is to transmit a call acknowledge packet, as illustrated by message 9D of FIG. 17A. The packet has the data content illustrated in Table 6. Both the and packets contain the information of the WebPhone users sending the packet. This information is useful by the recipient of the packet for a number of purposes. For example, the user information is displayed on the enunciator area of the WebPhone graphic display to identify the party placing the call. Second, the user may select such information and, using the drag and drop functionality of the WebPhone graphic user interface, add the user information to the callee WebPhone user's personal directory resident within his/her specific WebPhone application. In such a manner, both parties are completely identified to each other prior to commencing audio communications. The transmission of complete caller identification information with the and symbols packets enables such functions as individual or group specific call blocking, party specific outgoing messages, visual caller identification, and party specific priority ringing and sound effects, as explained herein.

Detailed Description Text (120):
Following transmission of packet by callee WebPhone 1538, the callee WebPhone further transmits an packet to caller WebPhone 1536, as illustrated by message 10 of FIG. 17A. Like the packet, the packet is essentially empty, containing nothing more than a session ID number which is unique to the call. The socket previously opened by caller WebPhone 1536 over which the forgoing packets were transmitted remains open for the transmission of control information between caller WebPhone 1536 and callee WebPhone 1538. Such control information may comprise an packet signaling the end of a call, a packet indicating that one of the parties to a call has placed the call "on hold" or other packets related to advance functionality of the WebPhone architecture. In addition, caller WebPhone 1536 opens a second socket to callee WebPhone 1538 over which the respective WebPhones may exchange packets, as illustrated by messages 11A-B of FIG. 17A. The packets have the data content illustrated in Table 6. The WebPhone application enables the parties to converse in real-time, telephone quality, encrypted audio communication over the Internet and other TCP/IP based networks. If both WebPhone client processes are utilized with full duplex sound cards, such as that illustrated in FIG. 12, the WebPhone users may transmit and receive audio packets simultaneously, similar to normal telephone conversation. However, if the WebPhone client processes are used with half duplex sound cards, a WebPhone user may only transmit or receive audio data simultaneously, similar to a speaker phone. Exchange of packets continues until either the callee WebPhone or the caller WebPhone transmits an packet, as illustrated by message 12 of FIG. 16A. Following the receipt of an end packet, the WebPhone client process will cease to accept subsequent audio packets.

Detailed Description Text (122):
Referring now FIG. 17B, if a WebPhone caller seeks to determine whether a prospective WebPhone callee is connected to the computer network, but, has little information regarding the client process, information server 1514 may be utilized as described. The WebPhone user defines One or more of the first name, last name, company, city, state, or country values of the Query field contained within the packet sends the packet to the global server. WebPhone 1536 opens a socket to global server 1500 and forwards packet to global server 1500, as illustrated by message 1 of FIG. 17B. Information server 1514 extracts the values specified the query field of the packet and queries the database 1516, as previously described with reference to FIG. 16B. Global server 1500 then transmits a packet back to WebPhone 1536, as

illustrated by message 2 of FIG. 17B. The packet has the format and data elements
indicated in Table 7, including the number of parties satisfying the search
criteria, specified in the packet. If the user of WebPhone 1536 wishes to receive
the number of parties satisfying the search criteria global server 1500
automatically transmits to WebPhone 1536 one or more packets, as illustrated by
messages 3A-C of FIG. 17B. The packet has the format and data elements as described
in Tables 6-7. At any time following transmission of the packet, WebPhone 1536 may
transmit an packet to either prevent transmission of any packets or to stop
transmission of any remaining packets, as illustrated by message 4 of FIG. 17B. The
packet has the format and data elements as described in Table 6-7.

Detailed Description Text (124):
The methods and apparatus described herein provide computer users with a powerful
protocol in which to directly establish real-time, point-to-point communications
over computer networks directly without server required linking. The a directory
server assists in furnishing the current dynamically assigned internet protocol
address of other similarly equipped computer users or information about such users.

Detailed Description Text (126):
In accordance with another aspect of the present invention, a system and method for
monitoring which of a plurality of WebPhone client processes are currently connected
to a computer network utilizes a server which dynamically adjust the required
notification interval of the WebPhone client processes depending on the utilization
of server resources.

Detailed Description Text (128):
Following the initial online notification sequence, as previously described with
reference to message 1-5 of FIG. 17A, WebPhone 1536 records the time at which the
packet was received from global server 1500. Simultaneously, global server 1500 sets
the "OnlineDate" field in the record for the WebPhone client process in Table 1516B.
The OnlineDate field has the form and data type as illustrated in Table 2.

Detailed Description Text (129):
Referring to FIG. 18, a schematic block diagram illustrating a packet transfer
sequence in accordance With the present aspect of the invention is illustrated.
Specifically, WebPhone 1536 transmit a packet to global server 1500 as illustrated
by message 1 of FIG. 18, in a manner as previously described with regard to message
1 of FIG. 17A. Global server 1500 responds with a packet as illustrated by message 2
of FIG. 18, in a manner as previously described with regard to message 2 of FIG.
17A. Following the receipt of the packet from global server 1500, WebPhone 1536 may
transmit any number of packets to the global server which will serve as notification
to the server that WebPhone 1536 is still actively coupled to the computer network.
As illustrated in FIG. 17B, such packets may include a packet, as illustrated by
message by 3A, a packet, as illustrated by message 3B or a packet, as illustrated by
message 3C, all of FIG. 18. The packet and the packet have been previously described
with reference to FIGS. 16A-B and FIGS. 17A-B.

Detailed Description Text (130):
The packet has the format and data content as illustrated in Table 7, and is
utilized by WebPhone server 1536 to notify global server 1500 that the WebPhone is
still actively connected to the network. The packet is issued after predetermined
periods of inactivity by WebPhone 1536. In response, global server 1500 updates the
"OnlineDate" field in the appropriate record of Table 1516B to reflect recent
activity from WebPhone client 1536. If the demand for server resources, i.e. the
number of transmissions to global server 1500, exceeds a predetermined threshold,
global server 1500 transmits to WebPhone 1536 and packet, as illustrated by message
4 of FIG. 18. The packet has the format and data content as illustrated in Table 7.
In the illustrative embodiment the value of the "AdjPulse" field within the packet
represents an offset to the default periodic interval value within WebPhone 1536.
Alternatively, the same field may represent an absolute time interval at which
WebPhone 1536 notifies global server 1500 of its status. In such instance, WebPhone
1536 will use the value contained in the packet as a temporary default until the
WebPhone logs off from the computer network or receives a new packet. If there is no
further interaction between Webphone 1536 and global server 1500, WebPhone 1536 will
send a subsequent packet to global server 1500 after expiration of the modified time

interval as defined in the prior received packet, as illustrated by <u>message</u> 5 of FIG. 18. If the resource demand on global server 1500 has increased or decreased since the last packet transmitted from WebPhone 1536, global server 1500 will update the on-line Table 1516B and select a new value for the "Adjpulse" field to be transmitted with the packet to WebPhone 1536. The value of the field may reflect a decrease in the default periodic notification interval, i.e., indicating a reduction in server resource utilization, or, conversely, an additional increase, i.e., an even larger utilization of server resource utilization. Global server 1500 will transmit the packet to WebPhone 1536, as illustrated by <u>message</u> 6 of FIG. 18. WebPhone 1536 will utilize the value contained within the packet and, at the appropriate interval, in the absence of further interaction with global server 1500, transmit another packet, as illustrated by <u>message</u> 7 of FIG. 18.

Detailed Description Text (131):
Referring to FIG. 19, a flowchart of the process by which global server 1500 dynamically determines whether to adjust the default period notification interval for client WebPhone processes currently connected to the <u>network</u> is illustrated. Global server 1500 and its subsystems, including connection server 1512, information server 1514, and database server 1518, remain in an idle state or attend to other tasks not related to the subject matter of this invention until a request from a WebPhone client process, such as WebPhone 1536 is received, as illustrated by decisional block 1910 of FIG. 19. Such a request may be in the form of any of the following packets: , , , , , , , , , and . Upon receipt of any of the previously described packets, global server 1500 increments the value of a Request Count variable maintained by global server 1500, as illustrated by process block 1912. In addition to the Request Count variable, global server 1500 also maintains a Threshold variable, both of which may be implemented with integer values. Global server 1500 next compares the value of the Request Count variable to the Threshold variable to determine whether or not the number of requests exceeds a previously defined level, as illustrated by decisional block 1914. If the current value of the Request Count variable exceeds that of the Threshold variable, global server 1500 will select a new value for the "Adjpulse" field to be sent with the packet to the requesting WebPhone 1536, as illustrated by procedural block 1916. The process of selecting a new value for the Adjpulse field may be implemented in global server 1500 by computing a Difference value between the Request Count variable and the Threshold variable and comparing the Difference value to a table of predefined Difference values, each having associated therewith a distinct value for the Adjpulse field. By successively comparing the Difference value with the predetermined values in the table, typically using less than or greater than comparisons, a new and appropriate Adjpulse value may be selected. The data structures necessary to implement such a table, as well as the code necessary to perform such calculations and comparisons are within the scope of those skilled in the programming arts and will not be defined in greater detail hereinafter. Further, global server 1500 transmits to WebPhone 1536 an packet containing the new value of the Adjpulse field, in a manner similar to that described with reference to <u>message</u> 4 of FIG. 18.

Detailed Description Text (134):
In the manner described above with reference to FIGS. 17B-19, the global server is able to accurately track which WebPhone client processes are active and currently coupled to the computer <u>network,</u> despite increased server activity. It will be obvious to those skilled in the arts that the actual implementation of the process described with reference to FIG. 19 may be embodied in connection server 1512, information server 1514, database server 1518, or, for the system described in FIG. 15B, as a separate server system entirely.

Detailed Description Text (135):
In an alternate embodiment, the various aspects of the invention may be implemented as a computer program product for use with a computer system. Such implementation may comprise a series of computer instructions either fixed on a tangible medium, such as a computer readable media, e.g. diskette 1142, CD-ROM 1147, ROM 1115, or fixed disk 1152 of FIG. 11, or transmittable to a computer system, via a modem or other interface device, such as communications adapter 1190 connected to the <u>network</u> 1195 over a medium 1191. Medium 1191 can be either a tangible medium, including but not limited to optical or analog communications lines, or may be implemented with

wireless techniques, including but not limited to microwave, infrared or other
transmission techniques. The series of computer instructions embodies all or part of
the functionality previously described herein with respect to the invention. Those
skilled in the art will appreciate that such computer instructions can be written in
a number of programming languages for use with many computer architectures or
operating systems. Further, such instructions may be stored using any memory
technology, present or future, including, but not limited to, semiconductor,
magnetic, optical or other memory devices, or transmitted using any communications
technology, present or future, including but not limited to optical, infrared,
microwave, or other transmission technologies. It is contemplated that such a
computer program product may be distributed as a removable media with accompanying
printed or electronic documentation, e.g., shrink wrapped software, preloaded with a
computer system, e.g., on system ROM or fixed disk, or distributed from a server or
electronic bulletin board over a network, e.g., the Internet or World Wide Web.

Detailed Description Paragraph Table (3):
TABLE 3 WebBoard Table Field Data Type Comments id ulong Unique ID Sequence image
varchar(255) Filename of image file imageType char GIF = 0, JPG = 1, RLE = 3 audio
varchar(255) Filename of TSP encoded.WAV file audioType char GSM = 0, TRUESPEECH = 1
hits ulong Number of accrued hits hitsprofile varchar(8) Filename of Demographics
version ulong version of WebBoard URL varchar(255) home page url

Detailed Description Paragraph Table (5):
TABLE 5 Source Table Field Data Type Comments id ulong Unique ID Sequence weboardID
ulong Link to WebBoard record name varchar(50) Company's name url varchar(80) URL to
Home Page street varchar(50) apt varchar(5) city varchar(20) state varchar(20)
country varchar(20) postalCode varchar(20) phone varchar(25) fax varchar(25) contact
varchar(35) Name of contact

Detailed Description Paragraph Table (8):
TABLE 8 WebPhone Protocol (WPP) Packet Data Definitions Element Data Type Comment
WPP_* unsigned char WPP message identifier sid unsigned long session id unique per
call version unsigned(3) version of the webphone (capability, protocol, vendor)
emailAddr varchar(90) email address of caller IPAddr varchar(80) IP Address
onlineState unsigned char bit 0 (ACTIVE/INACTIVE) bit 1 (Merchant Phone) bit 2
(Connection Server) bit 3 (webboard disabled) bit 4 Not Used bit 5 Not Used bit 6
Not Used bit 7 Not Used call Type unsigned char call type 0: EMAIL/1:IPCALL
partyEmailAddr varchar(90) email address of person to call connectStatus unsigned
char 0:NO WEBPHONE 1:ONLINE 2:OFFLINE 3:RECONNECT 4:PERM_RECONNECT partyIPAddr
varchar(80) IP Address of person to call userInfo varchar(120) firstName, LastName,
alias, emailAddr, street, apt, city, state, country, postalCode, phone, fax, company
audioType unsigned char audio compress type 0:GSM 1:TRUESPEECH

Detailed Description Paragraph Table (9):
TABLE 9 WebPhone Protocol (WPP) Packet Data Definitions Element Data Type Comment
length unsigned short length of audio or data in bytes audioData 512 Bytes
compressed audio data feature bits unsigned long WebPhone feature definition
fileType unsigned char file type 0:DATA 1:EMAIL 2:TEXT 3:BINARY fileName varchar(13)
name of file to be transmitted. fileSize unsigned long size of file to be
transmitted in bytes fileData variable file data query varchar(120) firstName,
lastName, company, city, state, country nparties unsigned long number of parties or
query records being sent size unsigned long size of file (IMAGE or AUDIO) to be sent
imageType unsigned char image type 0:GIF 1:JPG imageData 512 Bytes image data
eemailAddr varchar(90) encrypted email Address onlineStatus unsigned char 0 OK -1
Error callerStatus unsigned char 0 is unpaid 1 if paid onlineState unsigned char bit
0 webboard disabled bit 1 Not Used bit 2 Not Used bit 3 Not Used bit 4 Not Used bit
5 Not Used bit 6 Not Used bit 7 Not Used WBid unsigned long link to WebBoard record
adjpulse unsigned long timer offset in secs

Detailed Description Paragraph Table (11):
TABLE 11 Offset Name Size Description Reserved Reserved +1 SessionID 4 Unique value
for duration of this connection +5 Version 6 WebPhone version and distributor stamp
+11 Codec 1 Audio compression algorithm selected +12 FirstName 10 Given name, middle
initial +22 LastName 25 Surname +47 Alias 20 Nickname +67 Email- 90 Caller's
electronic mail address Addr +157 IpAddr 80 Caller's WebPhone's Internet address

+237 Street 50 Street address of user +287 Apt 20 Apartment or suite number +307
City 20 City name +327 State 20 State or province +347 Country 20 Country name +367
ZipCode 20 Zip or postal code +387 Phone 25 Telephone number +412 Fax 25 Facsimile
telephone number +437 Company 25 Employer or organization name +487 File Name 25
Name of file +512 Action 25 Action descriptor Code +537 File Type 10 File type
descriptor +547 Status 25 Status of WebPhone utility

CLAIMS:

1. In a server process capable of responding to requests from one or more client
processes operatively coupled to the server process over a computer network, a
method of monitoring the status of a client process, comprising the steps of:

A. receiving notification from a client process that the client process is active;

B. determining that the client process is inactive if a subsequent notification from
the client process is not received within a predetermined time interval;

C. monitoring a number of requests from a plurality of active client processes
coupled to the server over the computer network; and

D. increasing the predetermined time interval if the number of requests from the
plurality of active client processes at least equals a predetermined threshold.

9. A computer program product for use with a computer server operatively coupled to
one or more client processes over a computer network, the server capable of
responding to requests from the client processes, the computer program product
comprising a computer useable medium having program code embodied therein
comprising:

program code for receiving notification from a client process that the client
process is active;

program code for determining that the client process is inactive if a subsequent
notification is not received within a predetermined time interval;

program code for monitoring a number of requests from a plurality of client
processes operatively coupled to the server; and

program code for modifying the predetermined time interval, if the number of
requests from the plurality of client processes at least equals a predetermined
threshold.

10. The computer program product of claim 9 wherein the program code for determining
that the client process is inactive further comprises:

program code, responsive to the modified predetermined time interval, for
transmitting the modified predetermined time interval to active client processes
over the computer network.

12. A computer program product for use with a computer system operatively coupled to
a server process over a computer network, the computer system capable of executing
one or more client processes, the computer program product comprising a computer
useable medium having program code embodied therein comprising:

program code for connecting to the server process over the computer network;

program code for receiving from the server process data defining a predetermined
time interval within which to notifying the server process that a client process is
active;

program code for notifying the server process within the predetermined time interval
that a client process is active; and

program code for subsequently notifying the server process at predetermined time

intervals that the client process is still active; and program code for receiving a modified predetermined time interval from the server process.

15. The computer data signal of claim 14 wherein the program code for determining that the client process is inactive further comprises:

program code, responsive to the modified predetermined time interval, for transmitting the modified predetermined time interval to active client processes over the computer network.